

What is claimed is:

1. A method of providing an alternative view of a group of open windows on a display for a graphical user interface, comprising the steps of:
in response to a command to present the alternative view, repositioning all open windows of said group so that they appear in a respective area of the display without overlap, while maintaining the relative sizes and configurations of the windows in said group; and
subsequently returning the windows to their original positions in response to a user action.
2. The method of claim 1, wherein said user action is selection of one of the windows.
3. The method of claim 1, wherein said user action is issuance of a second command.
4. The method of claim 1 further including the step of resizing the windows so that all of the windows of said group appear within a defined boundary area of the display in the alternative view.
5. The method of claim 4 wherein all of the repositioned windows are resized according to a common factor to maintain their relative sizes and aspect ratios.
6. The method of claim 1 wherein said group comprises all open primary windows on the display.

7. The method of claim 6 further including the step of repositioning a subset of all open windows, which are related to one another, in response to a second command to present a second alternative view.

8. The method of claim 7 wherein said related windows are associated with a common application program.

9. The method of claim 7 further including the step of moving all windows out of the area of said display in response to a third command to present a third alternative view.

10. The method of claim 6 further including the step of moving all windows out of the area of said display in response to a second command to present a second alternative view.

11. The method of claim 1 wherein said group comprises a subset of all open windows on the display, which are related to one another.

12. The method of claim 11 wherein said related windows are associated with a common application program.

13. The method of claim 12 further including the step of repositioning and displaying a different subset of windows that are associated with a different application program in response to a predetermined command issued while in the alternative view.

14. The method of claim 1 further including the steps of:

detecting a user action indicating selection of one of said repositioned windows, in the alternative view; and
displaying the selected window in the foreground of the display upon returning the windows to their original positions.

15. The method of claim 1 wherein said windows are repositioned in a manner to maintain their relative positions.

16. The method of claim 15 wherein said repositioning step includes the steps of:

establishing a vector that indicates the relative positions of two overlapping windows, and

determining directions of movement for said overlapping windows in accordance with said vector.

17. The method of claim 16 wherein said windows are repositioned in an iterative manner, with a direction and amount of movement determined during each iteration.

18. The method of claim 17 wherein each iteration comprises the following steps:

determining a direction and amount of movement for each window to be repositioned;

scaling the movement amounts by a moderating factor; and

moving the windows by the scaled amounts.

19. The method of claim 18 wherein the moderating factor is varied for respective iterations.

20. The method of claim 18 further including the step of computing an external force factor to maintain the windows within a boundary area, and adding said force factor to the movement amounts.

21. The method of claim 17 further including the steps of determining whether the windows lie outside of a defined boundary area after the last iteration, and resizing the windows to fit within said boundary area if they lie outside of the boundary area.

22. The method of claim 1 further including the step of adding a border region to each window being repositioned, and repositioning the windows such that the border regions of the windows do not overlap.

23. The method of claim 1 wherein said command is initiated by user action.

24. The method of claim 23 wherein said user action is actuation of a physical element of a computer system.

25. The method of claim 23 wherein said user action is positioning of a cursor in a predetermined area of the display.

26. The method of claim 25 wherein said area is a predetermined corner of the display.

27. The method of claim 1 wherein said command is issued by a program.

28. The method of claim 27 wherein said program issues the command in response to detection of a specified condition.

29. The method of claim 1 wherein said repositioning step comprises moving said windows from their original positions to said respective areas over a discernable period of time to create an animation effect.

30. A computer system, comprising:

a display device;

a graphical user interface that normally displays a plurality of objects in a layered view in which an object can overlap and obscure at least a portion of another object;

means responsive to a command for repositioning said plurality of objects in an alternative viewing mode such that said objects appear in respective areas of the display device with the same relative sizes and configurations as in the layered view, but without overlapping any other objects of said plurality; and

means responsive to a subsequent command for returning said objects to their original positions in said layered view.

31. The computer system of claim 30, wherein said repositioning means comprises a component of an operating system program for said computer system.

32. The computer system of claim 30, wherein said repositioning means is contained in an application program that executes on said computer system.

33. The computer system of claim 30, wherein said repositioning means is contained in a plug-in module that cooperates with an operating system for said computer system.

34. The computer system of claim 30, wherein said objects comprise windows.

35. A graphical user interface for a computer having a first mode in which plural objects are displayed in a layered environment in positions in which an object can overlap and obscure at least some of the contents of another object, and a second mode in which said plural objects are temporarily moved from their positions in said first mode to respective areas within a display such that the content of each of said plural objects is visible without overlap, while maintaining the relative sizes and configurations of said plural objects, and subsequently returned to the position they occupied in said first mode.

36. The graphical user interface of claim 35 wherein said objects comprise windows.

37. The graphical user interface of claim 35 wherein said objects comprise a combination of windows and other user interface elements.

38. A program providing an alternative view for a computer user interface of the type that presents a normal viewing mode in which plural objects are displayed in a layered environment in positions in which an object can overlap and obscure at least some of the contents of another object, wherein said program causes said plural objects to move from their positions in said normal viewing mode to respective areas within a display such that the content of each of said plural objects is visible without overlap in said alternative viewing mode, while maintaining the relative sizes and configurations of said plural objects, and then return to their positions in the normal viewing mode.

39. The program of claim 38 wherein said objects comprise windows.

40. The program of claim 38 wherein said objects comprise a combination of windows and other user interface elements.

41. A computer-readable medium containing a computer program that is responsive to a predetermined command to execute a sequence of steps that animate overlapping objects on a display to move to respective areas of the display over a discernable period of time such that the content of each of said objects is visible without overlap of any of said objects at the end of said period, and responsive to a subsequent command to return the objects to their overlapping position.

42. The program of claim 41 wherein said objects comprise windows.

43. The program of claim 41 wherein said objects comprise a combination of windows and other user interface elements.

44. A method of providing an alternative view of a plurality of overlapping objects on a display for a graphical user interface, comprising the steps of:

in response to a command to present the alternative view, moving the objects to respective positions on the display so that they appear without overlap;
and

returning the objects to their original positions in response to a user action.

45. The method of claim 44 wherein the relative sizes of said objects are maintained during said movement and while they are located at said respective positions.

46. The method of claim 45, wherein said user action is selection of one of the objects.

47. The method of claim 45, wherein said user action is issuance of a second command.

48. The method of claim 44 further including the step of resizing the objects so that all of the displayed objects appear within a defined boundary area of the display in the alternative view.

49. The method of claim 48 wherein all of the repositioned objects are resized according to a common factor to maintain their relative sizes.

50. The method of claim 44 wherein said objects include icons.

51. The method of claim 44 wherein said objects include images.

52. The method of claim 44 wherein said objects are repositioned in a manner to maintain their relative positions.

53. A method of displaying windows in a user interface for a computer, comprising the steps of:

displaying windows in a layered view where at least one window can overlay another window and obscure at least a portion of the contents of said other window;

switching to an unlayered view in response to a first command, wherein a predetermined set of windows are displayed without overlap while maintaining their relative sizes and configurations; and

returning to said layered view in response to a second command.

54. The method of claim 53 wherein said set of windows comprises all open windows being displayed.

55. The method of claim 53 wherein said set consists of all open windows associated with one application program.

56. The method of claim 53 further including the step of dragging an object from one of the windows in said set to another of the windows in said set while said unlayered view is being displayed.

57. A method for facilitating interactivity between objects appearing on a desktop and in windows of a computer user interface, comprising the steps of:

displaying one or more windows in a normal view such that the windows can obscure a user's view of objects on the desktop of the user interface;

temporarily removing the windows from their obscuring positions in response to a first user command;

selecting at least one of said desktop objects while the windows are removed;

returning the windows to their original positions in response to a second command from the user, while maintaining the selection of said desktop object; and

placing the selected object in one of said windows.

58. The method of claim 57 wherein the step of temporarily removing the windows comprises the steps of displaying a border area along at least one edge of the desktop, and moving the windows to positions within said border area.

59. The method of claim 58 wherein the step of returning the windows is initiated by dragging the selected desktop object to said border area.

60. A method for facilitating interactivity between objects appearing on a desktop and in windows of a computer user interface, comprising the steps of:

displaying one or more windows in a normal view such that the windows can obscure a user's view of objects on the desktop of the user interface;

selecting an object in a window;

temporarily removing the windows from their obscuring positions in response to a first user command, while maintaining the selection of the object;

placing the selected object on the desktop or a desktop object while the windows are removed; and

returning the windows to their original positions in response to a second command from the user.

61. A method for displaying windows in a graphical user interface for a computer, comprising the steps of:

displaying a plurality of windows that are respectively associated with different applications running on the computer;

in response to a command to present an alternative view, repositioning those windows associated with one of said applications so that they appear in a respective area of the display without overlap in the foreground of the display; and subsequently returning the windows to their original positions in response to a user action.

62. The method of claim 61, further including the step of dimming the appearance of the windows associated with the applications other than said one application.

63. The method of claim 61, wherein said one application is the application that is active when said command is issued.

64. The method of claim 61, further including the steps, in response to another user action during the time that said windows associated with said one application are repositioned, of:

repositioning those windows associated with second application so that they appear in a respective area of the display without overlap in the foreground of the display; and

dimming the windows associated with said one application.